

Gas Detection System

Aim

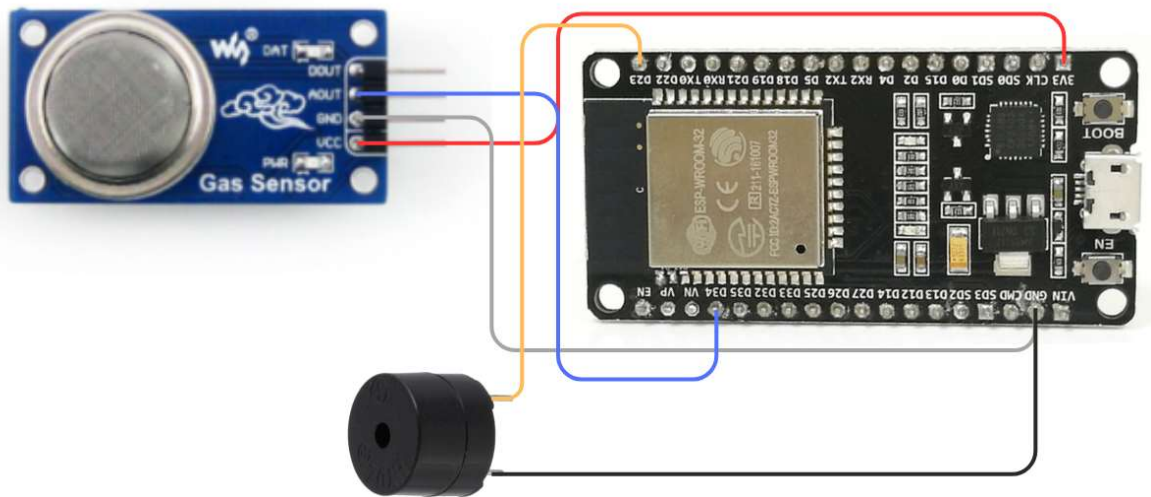
This project presents a gas detection system that utilizes an ESP32 microcontroller to monitor gas levels using an analog gas sensor. The system alerts users with a buzzer and LED when gas levels exceed a certain threshold. By connecting to Wi-Fi, it enables remote monitoring through the Blynk app, making it ideal for IoT applications.

Components Needed

COMPONENTS	CONTENT
ESP32 Development Board	1
Gas Sensor (MQ-2)	1
Buzzer	1
Jumper Wires	As required

Circuit Connections

COMPONENTS	ESP32 Pin	Description
Gas Sensor	GPIO34	Sensor data pin connected to GPIO 34; measures gas concentration.
Buzzer	GPIO 23	Activates when gas concentration is above the threshold.



Code Explanation

Libraries and Setup

1. **WiFi and Blynk Libraries:** These libraries are essential for enabling Wi-Fi and Blynk functionalities.
2. The Blynk authentication token, Wi-Fi SSID, and password are defined at the start for secure connection.

Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3YX0Qp2HY"
#define BLYNK_TEMPLATE_NAME "gas sensor"
#define BLYNK_AUTH_TOKEN "BSxx9fQ5YjSKC_lap_3CBv_0LfycDQIl"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "BSxx9fQ5YjSKC_lap_3CBv_0LfycDQIl";
char ssid[] = "Shivam";
char pass[] = "shivam2724";
```

```
#define sensor 34
#define buzzer 23

BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  pinMode(buzzer, OUTPUT);
  pinMode(sensor, INPUT);
}

void GASLevel() {
  int value = analogRead(sensor);
  value = map(value, 0, 4095, 0, 100);

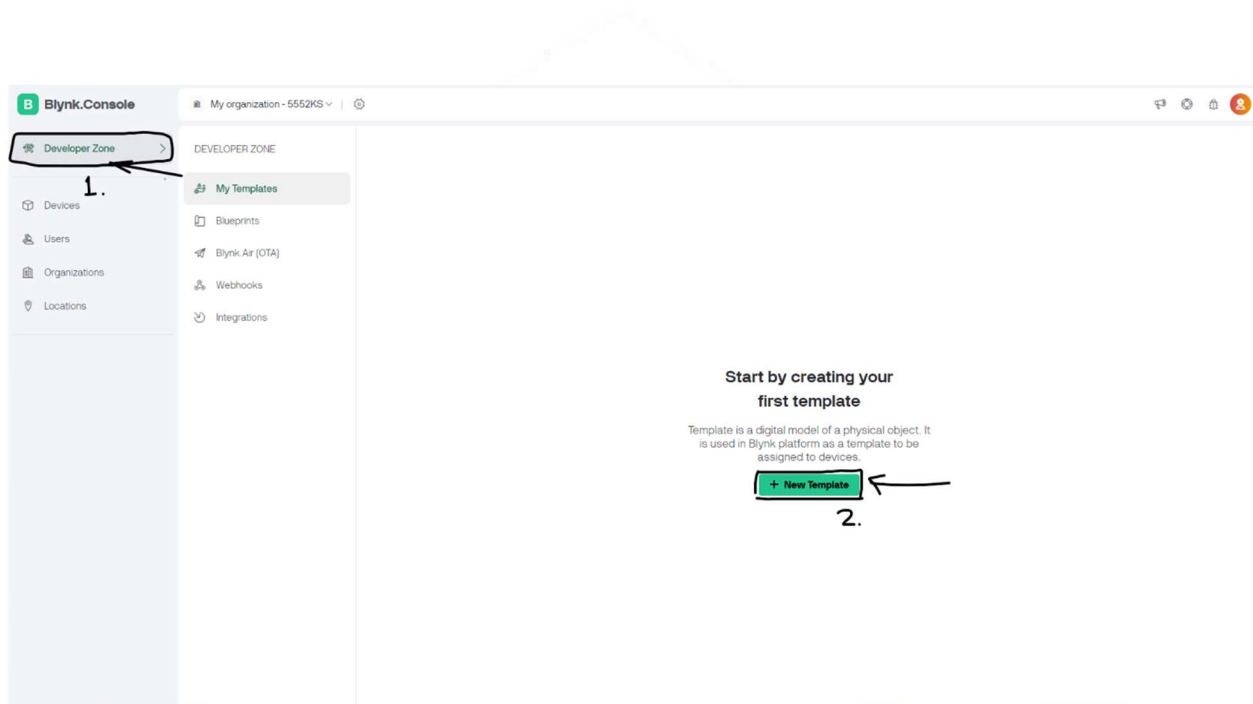
  if (value >= 25) {
    digitalWrite(buzzer, HIGH);
    WidgetLED LED(V1);
    LED.on();
  } else {
    digitalWrite(buzzer, LOW);
    WidgetLED LED(V1);
    LED.off();
  }

  Blynk.virtualWrite(V0, value);
  Serial.println(value);
}

void loop() {
  GASLevel();
  Blynk.run();
  delay(200);
}
```

Blynk App Setup

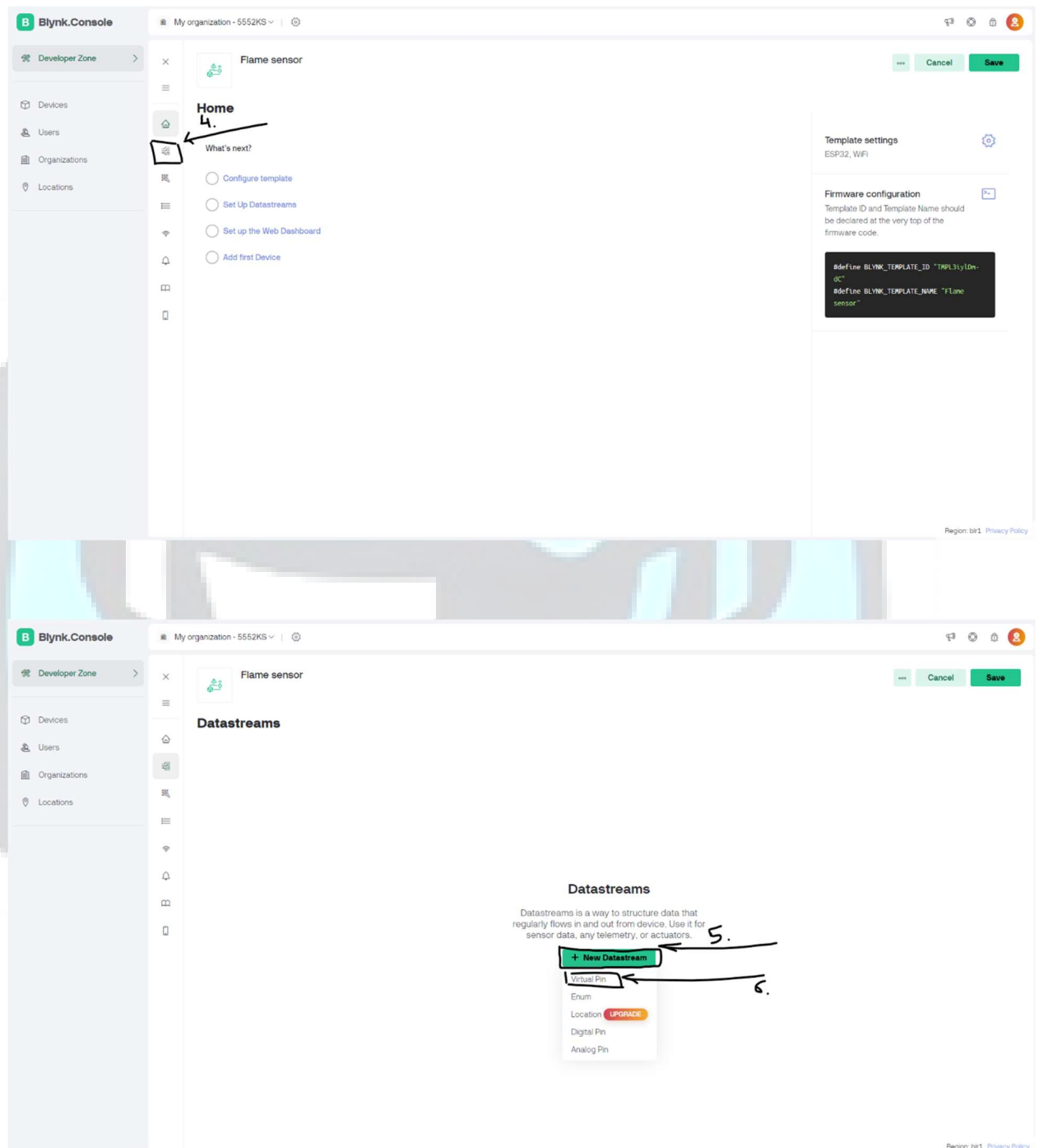
1. **Create a New Project** in the Blynk app with the template provided in the code.



2. Enter name of the New Template then click OK.

The screenshot shows the 'Create New Template' dialog box. The title is 'Create New Template'. There are three main sections: 'NAME', 'HARDWARE', and 'CONNECTION TYPE'. The 'NAME' section has a text input field containing 'Flame sensor' and a character count '12 / 50'. The 'HARDWARE' section has a dropdown menu showing 'ESP32'. The 'CONNECTION TYPE' section has a dropdown menu showing 'WiFi'. There is also a 'DESCRIPTION' section with a large text area containing the placeholder text 'Description' and a character count '0 / 128'. At the bottom right, there are two buttons: 'Cancel' and 'Done'.

3. Click on DATASTREAM tab then click on the new datastream and select Virtual Pin.



4. Write the Name then select the Virtual PIN according to the Code and click create button.

Flame sensor

Datastreams

Virtual Pin Datastream

General
Expose to Automations

NAME
Integer V0

ALIAS
Integer V0

PIN
V0

DATA TYPE
Integer

UNITS
None

MIN
0

MAX
1

DEFAULT VALUE
0

☐ Enable history data

Cancel
Create

5. Click on Web Dashboard Tab then click and drag the appropriate widget from Widget Box .

Blynk.Console

My organization - 5552KS

Developer Zone

Flame sensor

Web Dashboard

Widget Box

CONTROL

Switch

Slider

Number Input

Image Button

Web Page Image Button

Device Name Online

Device Owner Company Name

Dashboard

Switch

Gauge

1h 6h 1d 1w 1mo 3mo

Region: bkr1 Privacy Policy

6. Click on settings button on widget then choose the appropriate datastream then click save.

Web Dashboard

Widget Box

CONTROL

Switch

Slider

Number Input

Image Button

Web Page Image Button

Dashboard

Device Name Online

Device Owner Company Name

13.

Switch Settings

TITLE (OPTIONAL)

Switch

Datastream

Choose Datastream

+ Create Datastream

14.

Switch

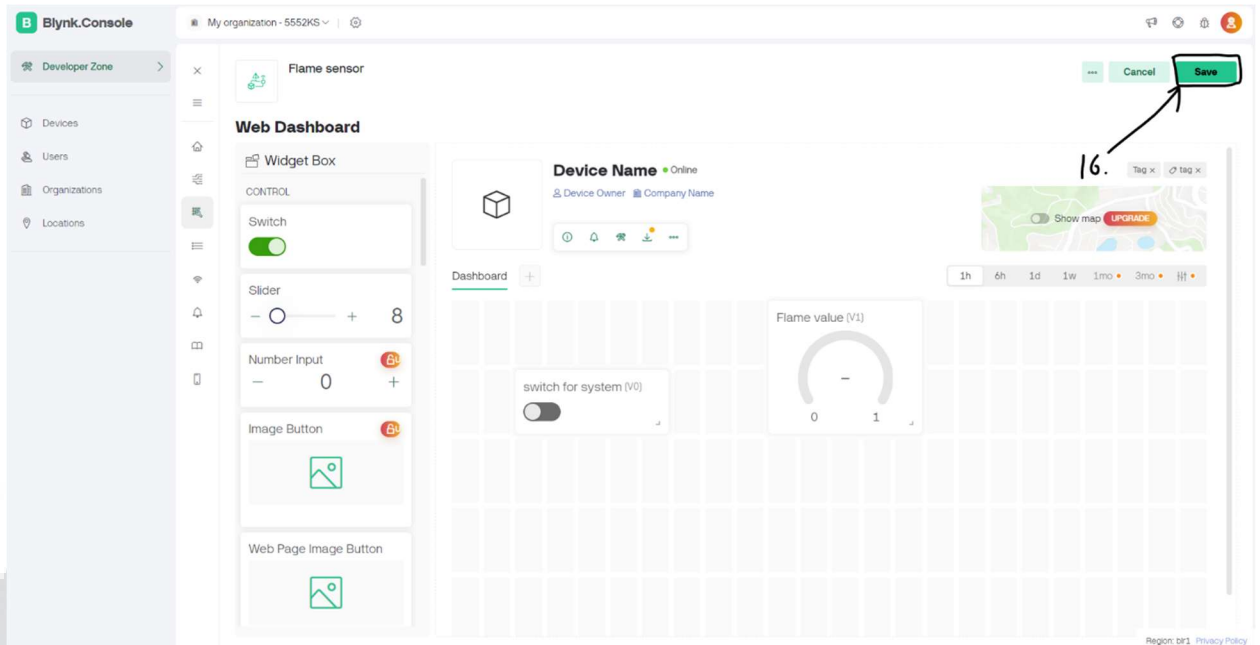


Cancel

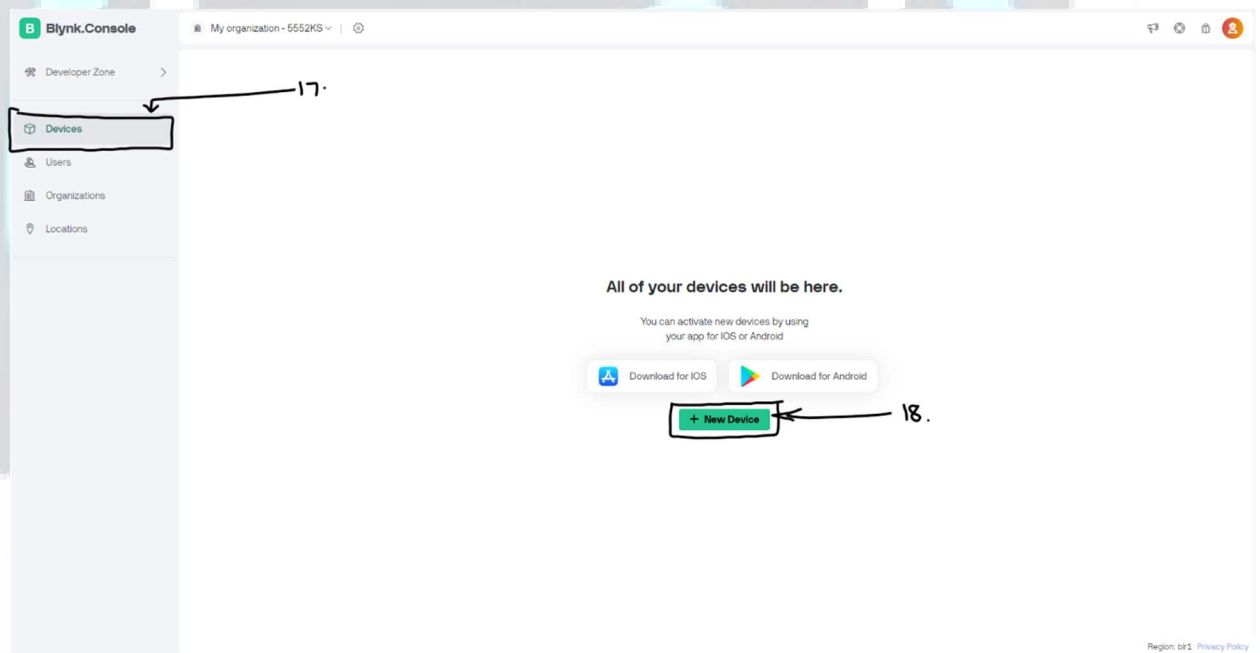
Save

15.

6. Click SAVE button.




7. Click on Devices Tab then click on NEW Device and click on "From template".




New Device

Choose a way to create new device


From template




Scan QR code



Manual entry



 Point on the cards to see instructions

Cancel

8. Choose the appropriate Template You are using then click on create button.

New Device

Create new device by filling the form below

TEMPLATE

Choose template

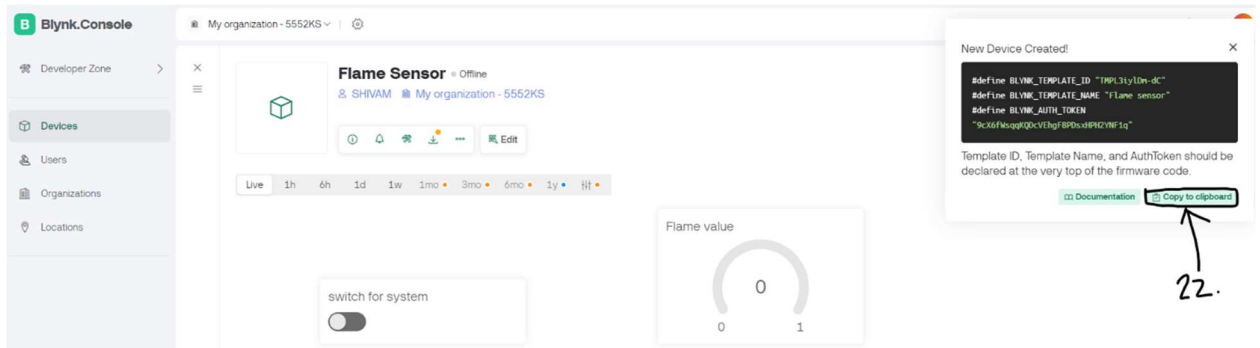
DEVICE NAME

New Device 0 / 50

Use letters, digits, apostrophe, underscore, hyphen and spaces only

Cancel Create

9. Click on copy to clipboard to copy the credentials.



10. Replace the copied credentials with the top part of credentials and change the authentication code according to your credentials and change the ssid with your wifi name and pass with your wifi password.

```
#define BLYNK_TEMPLATE_ID "TMPL3CtJJjB1C"
#define BLYNK_TEMPLATE_NAME "Fire alarm"
#define BLYNK_AUTH_TOKEN "oCSU7yexudWcyKDZsj1r-j88sdk5btac"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Blynk authentication token
char auth[] = "oCSU7yexudWcyKDZsj1r-j88sdk5btac"

// Wi-Fi credentials
char ssid[] = "Shivam";
char pass[] = "shivam2724";

// Define pins
#define LED1 12 // LED indicating fire detected
#define LED2 13 // LED indicating system is safe
#define Buzzer 23 // Buzzer
#define Sensor 2 // Flame sensor

BlynkTimer timer;
int pinValue = 0; // Variable to hold system state
int sensorValue = 0; // Variable to hold sensor state

// Create LCD object with I2C address 0x27 (adjust if necessary)
LiquidCrystal_I2C lcd(0x27, 16, 2);

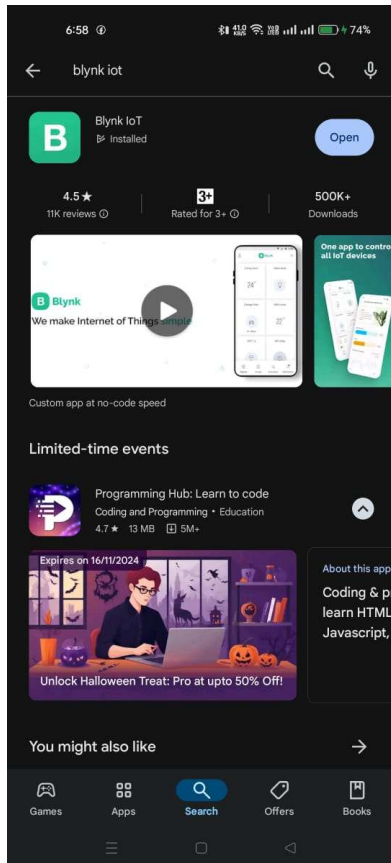
void setup() {
  Serial.begin(9600);

  // Initialize pins
  pinMode(LED1, OUTPUT);
```

12 . Then select the PORT in your Arduino IDE then Select the appropriate ESP32 model you are using (according the board we are using is ESP32 Devkit module then upload it.

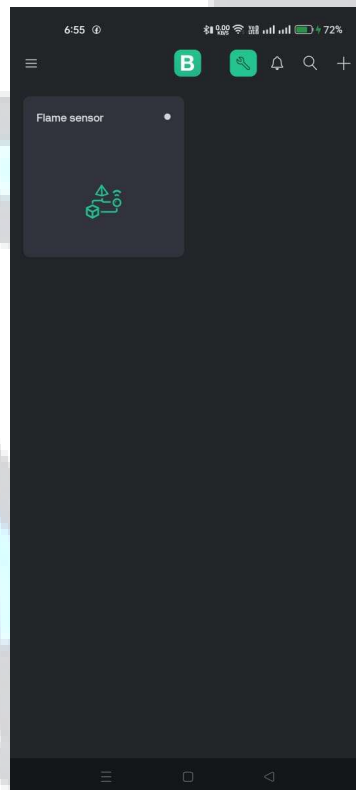
Setup for Blynk IoT Mobile

1. You have to download the Blynk IoT app from the Play Store or iPhone store .

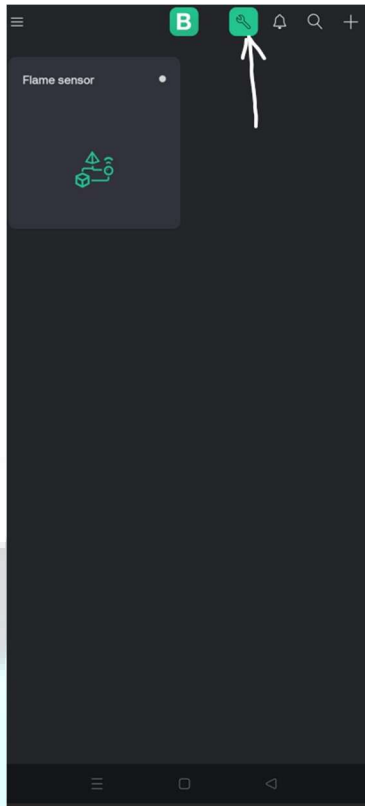


2. Then Login with the same ID and Password for which you have used for Blynk IoT in your PC web browser.

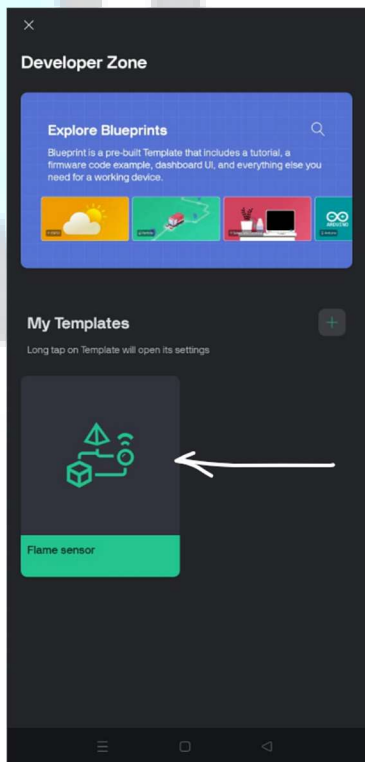
3. After Login , Your device will be already created in your phone as you have created in your IoT web browser.



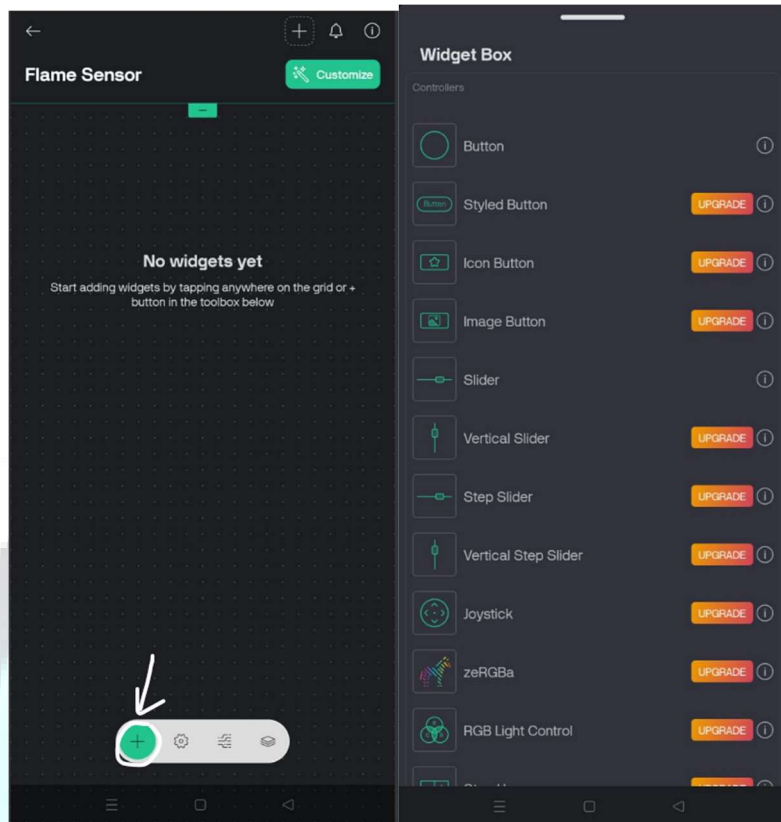
4. Click on Developers Tool Button.



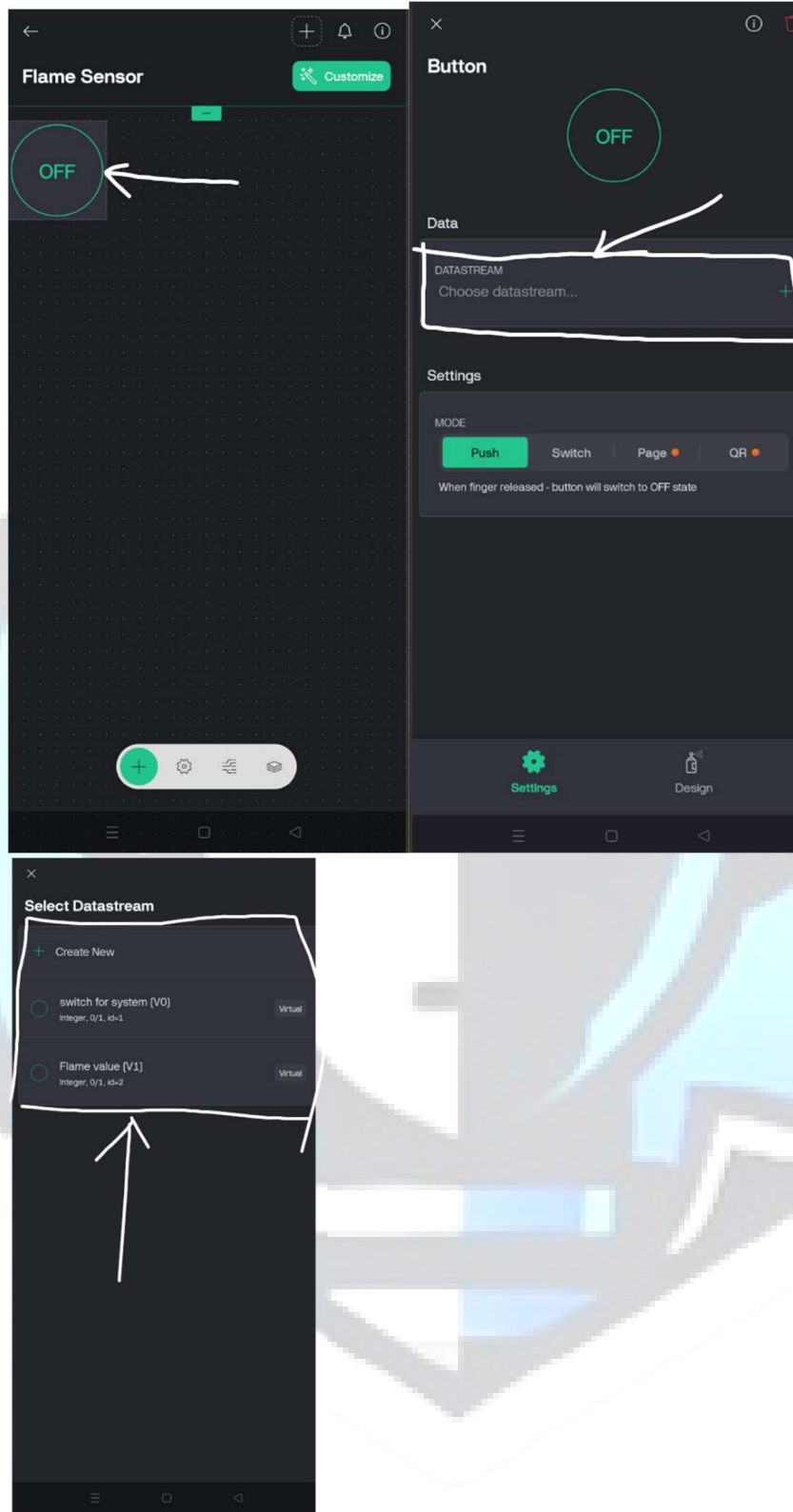
5. Click on Already Created template .



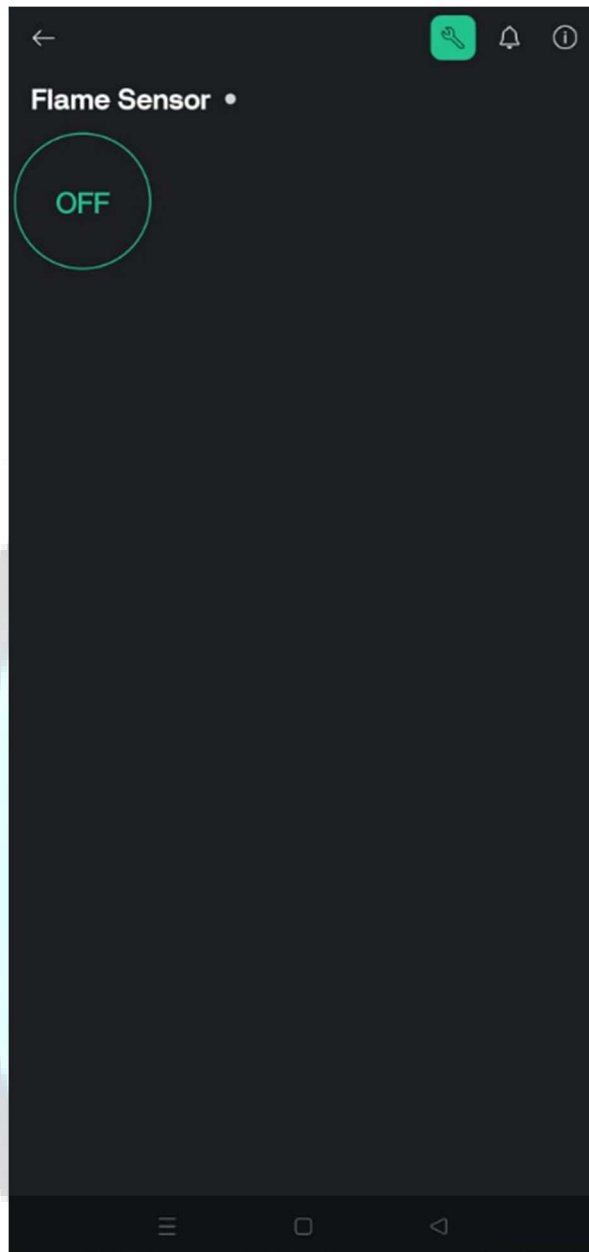
6. Click on + icon as shown in the figure to choose the appropriate widget from widget box.



7. Then click on the widget you are going to use and in settings option choose the appropriate datastream for that widget.



8. Then just go back to device . And you have created the IoT device in your phone also .



Conclusion

This gas detection project, using ESP32, Blynk IoT integration, and an analog gas sensor, provides an efficient way to monitor gas levels in real time. The system promptly activates a buzzer and visual alert when gas concentration surpasses a safe threshold. Through Blynk, it allows for remote monitoring, adding an extra layer of safety for IoT applications. This project is an excellent introduction to IoT-based safety solutions, helping students understand the integration of sensors and microcontrollers with cloud-based applications for effective monitoring and control.

